1982

Computer Aided Filter Design Using Intel SPAS20 Software

Robert L. Olive
University of Central Florida

COMPUTER AIDED FILTER DESIGN USING INTEL SPAS20 SOFTWARE

BY

ROBERT LYOD OLIVE, II.
B.S., Northern Arizona University, 1976
M.S., Northern Arizona University, 1978

RESEARCH REPORT

Submitted in partial fulfillment of the requirements
for the degree of Master of Science
in the Graduate Studies Program of the College of Engineering
University of Central Florida
Orlando, Florida

Summer Term
1982

ABSTRACT

This paper demonstrates conversion of an analog filter into a digital filter using computer aided software. The filter design to be demonstrated is a common third order Butterworth filter. This paper is not an attempt to review all filter designs or applications, but rather the attempt is to give a detailed explanation of the steps required to design almost any digital filter.

No knowledge of the Intel Series 210 microcomputer development system is assumed. The appendices contain introduction to the Series 210 system.

Chapter I demonstrates the steps needed to design this filter without computer aid. Included are both analog and digital filter response characteristics.

Chapter II supplemented with Appendix C demonstrates the computer aided filter design. Again, filter characteristics are included.

Chapter III compares the results of Chapters I and II.

Even though this paper attempts to be inclusive of most of the computer details, it should not be used in exclusion of the available Series 210 manuals.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

INTRODUCTION

Computer aided design is employed extensively in industry. There are many advantages to this approach to design. Increased speed and reduced cost would certainly be grouped together at the top of the list. There are, however, some pitfalls in computer aided design. One of these would be separation of the engineering from the engineer. As a result of this, Chapter I is devoted to completely designing the example filter from fundamental filter concepts and a knowledge of s- to z-plane transforms.

The Intel 2920 microcomputer was selected for this filter design hardware. There is a full complement of available software for complete computer aided design. The filter can be designed first in analog form with SPAS20.SFT, and the corresponding SPAS20.MC1 macro. Conversion of the s-plane poles and zeros to z-plane poles and zeros is accomplished with the SPAS20.MC2 macro. Finally, the poles and zeros can be realized in cascade, canonic form, assembly language, by using the SPAS20.MC3 macro. A variety of other canned macros are available under the SPAS. object code. These are listed in reference 1.

Once the filter has been programmed in assembly language, the completed program can be assembled using AS2920. Simulation can be performed using a variety of different inputs with SM2920.SFT.

CHAPTER I

CONVENTIONAL DIGITAL FILTER DESIGN

In this chapter, a common filter design problem is stated and solved completely using conventional and proven digital methods. The analog filter is first designed. Included are graphs of frequency response and step response. A transform method discussion is presented and the bilinear transform is used to transform from the s-plane to the z-plane. The z transfer function is put into standard form and partial fraction expanded in anticipation of the design realization. The poles and zeros are individually realized and the assembly language program is generated. Graphs of frequency response and step response for the z transfer function are presented. Finally, analog and digital responses are compared.

Problem Statement

Use the bilinear transform method to design a third order, low pass Butterworth filter with cutoff frequency of five hundred hertz and sample frequency of four thousand hertz.

Step One: This step consists of designing the corresponding third order, low pass, Butterworth analog filter. We begin by describing the loss function, $L(w^2)$, as equation 1:

$$L(w^2) \equiv 1 + w^{2N} \qquad (1)$$

where:

N = function order

w = S/j

S = s normalized

Substitution and factoring yields equation 2:

$$L(w^2) = (S + 1)(S - 1)(S^2 + S + 1)(S^2 - S + 1) \qquad (2)$$

Choose poles in the left half s-plane for stability, and define the gain function, H(S):

$$H(S) \equiv 1/(S^3 + 2S^2 + 2S + 1) \qquad (3)$$

This transfer function could also be derived from tables of coefficients listed in reference 2. A TI-59 program to produce gain in increments of fifty hertz, and the gain output are listed in Appendix A. Figure 1 is the frequency response of this transfer function. The step response is shown in Figure 2.

Step Two: One of the most popular methods to transform from the prototype analog filter to the digital filter is the bilinear transform. This transform process can be used effectively two different ways. The first employs a one for one mapping of a particular frequency, in this case the cutoff frequency, to the same frequency in the digital filter. Thus, the digital realization has the same cutoff frequency as the analog prototype filter. This is the primary reason for the bilinear transform popularity.

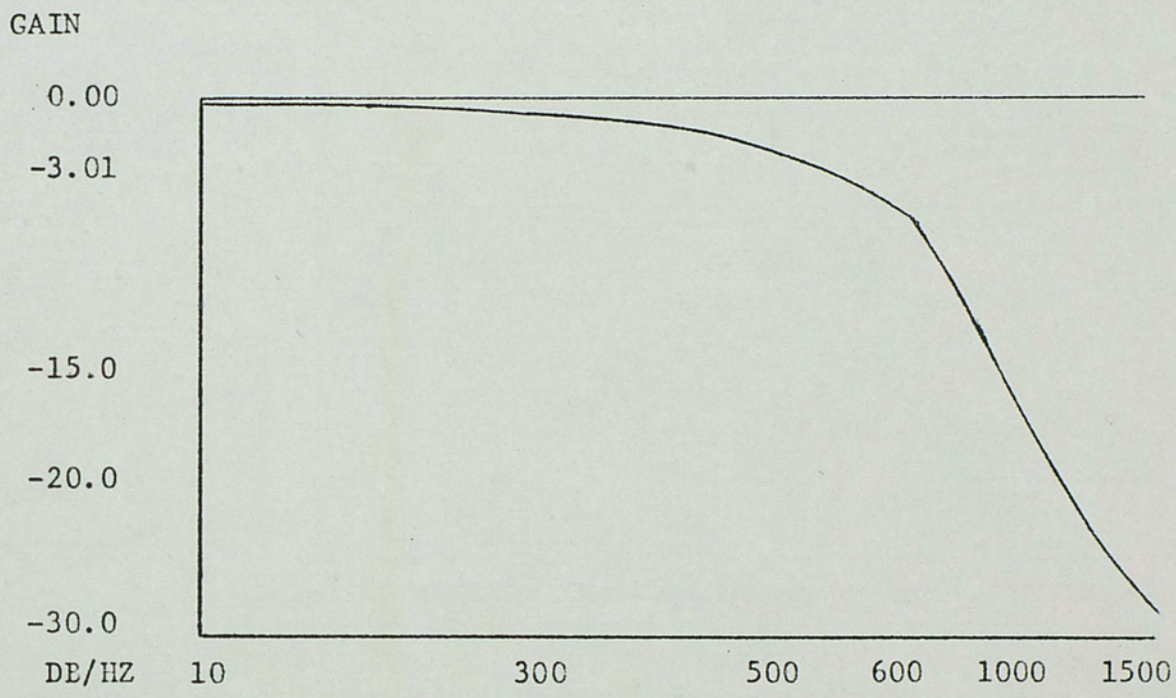

Fig. 1. The third order Butterworth frequency response.

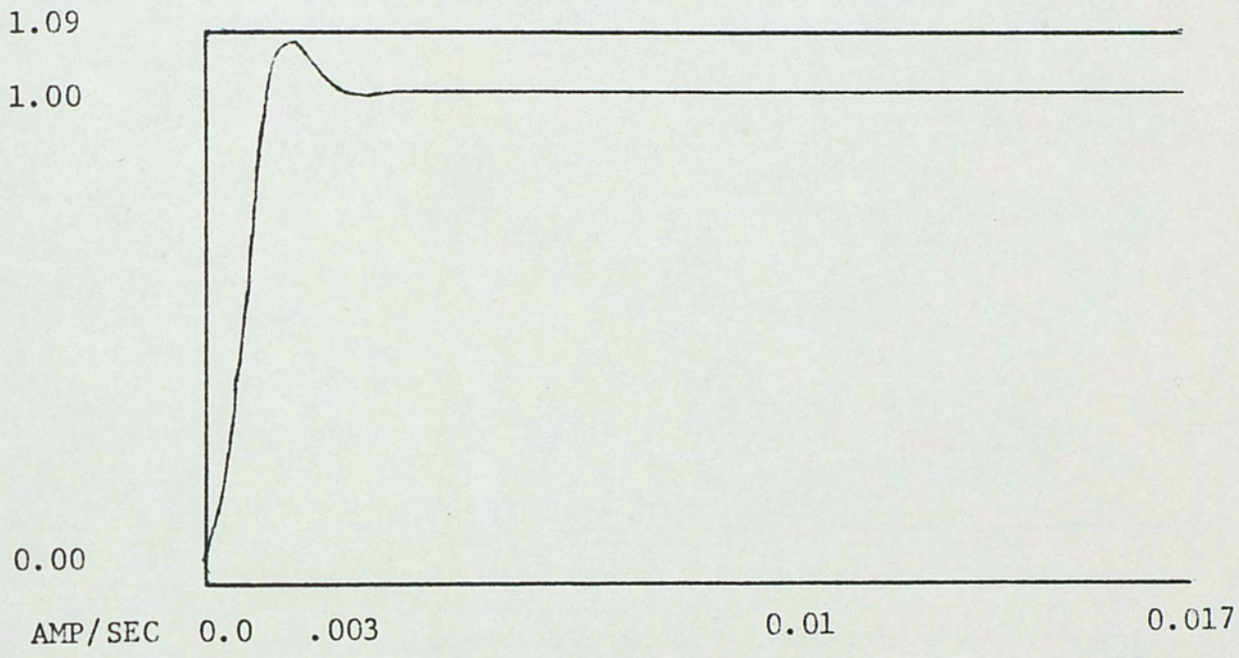

Fig. 2. The third order Butterworth step response.

The second method is suggested by the relationship:

$$\theta \doteq \tan \theta$$

for small values of θ. This is a direct consequence of the bilinear transform being a tangential relationship. In this method, the design criteria is simply to require a low frequency match with respect to the s-z plane mapping.

We now convert from s to z plane using the bilinear transform which matches cutoff frequency. See equation 4 and equation 5:

$$H(z) = H(S)_{S\leftarrow \frac{C(1 - z^{-1})}{(1 + z^{-1})}} \tag{4}$$

$$C = \cot(\pi f_{co}/2f_0) \tag{5}$$

where, f_0 is the sample frequency divided by two. This is sometimes referred to as the folding frequency. If we list the z transfer function in standard form (equation 6), evaluate constant C, we can employ reference 2 to evaluate the coefficients. The coefficients for the Butterworth filter are listed in Table 1.

$$H(z) = \frac{(a_0 + a_1 z^{-1} + a_2 z^{-2} + \ldots + a_n z^{-n})}{(1 + b_1 z^{-1} + b_2 z^{-2} + \ldots + b_n z^{-n})} \tag{6}$$

TABLE 1

Z-TRANSFER COEFFICIENTS AND CONSTANT C

C	2.4142135625
a_0	.03616893439
a_1	.0950680316
a_2	.0950680316
a_3	.0316893439
b_1	-1.459029062
b_2	.910369
b_3	- .1978251872

In anticipation of filter realization, we now partial fraction expand equation 6. In order to maintain low sensitivities, the filter transfer function needs to be realized in the smallest possible degree sections. This is a direct result of quantization error, accumulating roundoff error, and coefficient bit resolution error affects on pole, zero locations; which in turn affects the stability of the final filter. One should not attempt to partial fraction expand a transfer function where the order of the numerator is greater than or equal to the order of the denominator. One way around this difficulty is to divide the numerator by the denominator, thus generating a number and a remainder. The numerator

of the remainder will now be a lower order polynomial than the denominator. After division of equation 6, equations 7 and 8 are generated.

(7)

$$H(z) = .0316893439 + \frac{.1413037054z^2 + .0662190354z + .0379582943}{z^3 - 1.459029062z^2 + .910369z + .1978251872}$$

(8)

$$H(z) = .0316893439 + \frac{A}{(z - .4142135625)} + \frac{Bz + C}{(z^2 - 1.0448155 + .4775922499)}$$

Table 2 contains these coefficient values.

TABLE 2

PARTIAL FRACTION EXPANSION COEFFICIENTS

A = .4142135625
B = -.2729098574
C = .3859528211

Step Three: The transfer function of equation 8 is best suited to the parallel canonic form. This direct form will minimize the number of RAM locations needed to implement the filter. The parallel canonic form used, with the assigned RAM locations, is shown in Figure 3. The assigned multiplication coefficients are listed in Table 3 with the associated binary representations.

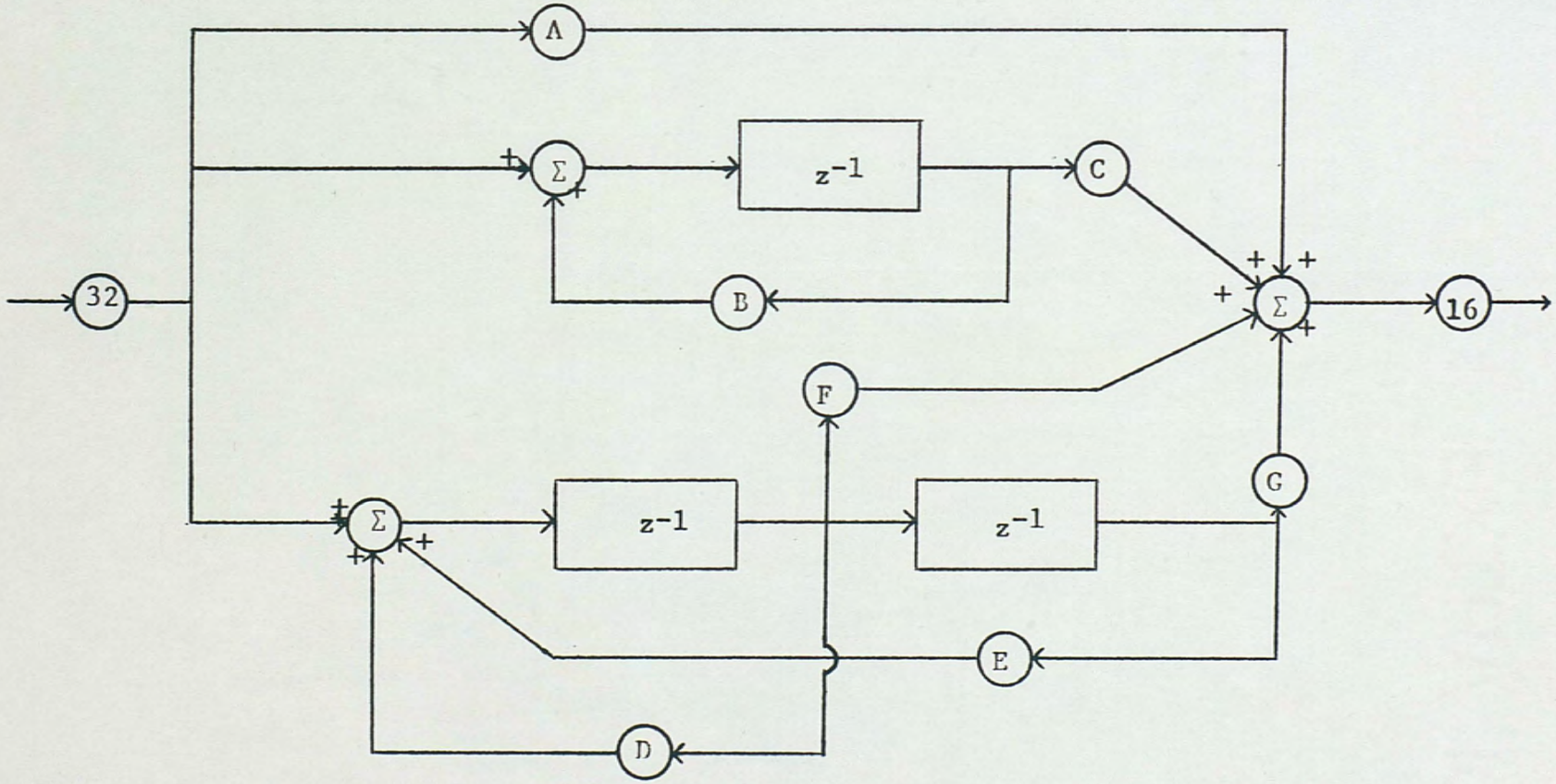

Fig. 3. Parallel canonic form for digital filter realization.

TABLE 3

BINARY REPRESENTATION

A =	.0316893439	$2^{-5}{}_{+2}{}^{-11}{}_{-2}{}^{-14}{}_{+2}{}^{-16}{}_{-2}{}^{-18}{}_{+2}{}^{-21}{}_{+2}{}^{-23}{}_{+2}{}^{-24}$
B =	.4142135625	$2^{-1}{}_{-2}{}^{-4}{}_{-2}{}^{-6}{}_{-2}{}^{-7}{}_{+2}{}^{-12}{}_{-2}{}^{-13}{}_{+2}{}^{-15}{}_{-2}{}^{-19}{}_{+2}{}^{-21}{}_{-2}{}^{-23}$
C =	.4142135625	$2^{-1}{}_{-2}{}^{-4}{}_{-2}{}^{-6}{}_{-2}{}^{-7}{}_{-2}{}^{+12}{}_{-2}{}^{-13}{}_{+2}{}^{-15}{}_{-2}{}^{-19}{}_{+2}{}^{-21}{}_{-2}{}^{-23}$
D =	1.0448155	$2^{0}{}_{-2}{}^{-5}{}_{-2}{}^{-6}{}_{-2}{}^{-9}{}_{-2}{}^{-13}{}_{+2}{}^{-16}{}_{+2}{}^{-21}{}_{-2}{}^{-25}$
E =	- .4775922499	$-2^{-1}{}_{+2}{}^{-6}{}_{+2}{}^{-7}{}_{-2}{}^{-10}{}_{-2}{}^{-14}{}_{+2}{}^{-17}{}_{+2}{}^{-22}$
F =	- .2729098574	$-2^{-2}{}_{-2}{}^{-6}{}_{-2}{}^{-7}{}_{+2}{}^{-11}{}_{+2}{}^{-15}{}_{+2}{}^{-17}{}_{+2}{}^{-20}{}_{+2}{}^{-22}{}_{+2}{}^{-25}$
G =	.3849528211	$2^{-1}{}_{-2}{}^{-3}{}_{+2}{}^{-7}{}_{-2}{}^{-8}{}_{-2}{}^{-10}{}_{+2}{}^{-12}{}_{-2}{}^{-15}{}_{-2}{}^{-18}{}_{+2}{}^{-20}{}_{-2}{}^{-23}$

For future comparison, the poles of the partial fractions in both cartesian and polar coordinates are given in Table 4.

TABLE 4

POLES AND ZEROS OF DIGITAL FILTER TRANSFER FUNCTION

Polar: Radius, angle

Poles	Zeros
.6910804945, .713724379	1.000000000, pi
.4142135625, 0.000000000	1.00000000, pi

Cartesian:

Poles

.5224077498, $\pm$ j .4524183825

.4142135625, 0.000000000

A note of caution is needed when referring to the parallel canonic form given in Figure 3. The feedback is easily shown to be negative. In assignment of the feedback coefficients, the minus sign has been absorbed into the coefficient.

Conversion of this filter into assembly language now follows. The number of right shifts which are permitted by the Intel 2920 microprocessor is thirteen. One technique of obtaining more resolution is to right shift the variable thirteen places and store it in a temporary location, TEMP, then right shift the temporary location the desired number of times up to eleven. Further assembly language details can be obtained from reference 3. A listing of the design example assembly program is given in Table 5. The computer generated printout, PAPA1.RLO, and the assembly generated listings are given in Appendix B. Appendix C is a discussion of getting on line with the computer, assembly and simulation procedures. Discussion of CREDIT is included also.

Step Four: This is the simulation step. Intel simulation software is available in the form of SM2920.SFT. A complete description of this simulation software is given in reference 4.

The expected filter response can be obtained from the z-transfer function. Converting the transfer function into a difference equation is straight forward. Equation 9 shows the associated difference equation. Table 6 contains tabulated output values for fifteen sample periods using a step input. The simulation output

TABLE 5

ASSEMBLY PROGRAM FOR THIRD ORDER BUTTERWORTH FILTER

		D*X1		
LDA	X1	X2	R00	IN1
ADD	X1	X2	R05	IN1
ADD	X1	X2	R06	IN1
SUB	X1	X2	R09	NOP
SUB	X1	X2	R13	NOP
LDA	TEMP	X2	R13	CVTS
ADD	X1	TEMP	R03	NOP
ADD	X1	TEMP	R08	NOP
SUB	X1	TEMP	R12	CVT7
SUB	X1	TEMP	R13	NOP
		E*X3		
SUB	X1	X3	R01	NOP
ADD	X1	X3	R06	CVT6
ADD	X1	X3	R07	NOP
SUB	X1	X3	R10	NOP
LDA	TEMP	X3	R13	CVT5
SUB	X1	TEMP	R01	NOP
ADD	X1	TEMP	R04	NOP
ADD	X1	TEMP	R09	CVT4
SUB	X1	TEMP	R13	NOP
		B*X5		
LDA	X4	X5	R01	NOP
SUB	X4	X5	R04	CVT3
SUB	X4	X5	R06	NOP
SUB	X4	X5	R07	NOP
ADD	X4	X5	R12	CVT2
SUB	X4	X5	R13	NOP
LDA	TEMP	X5	R13	NOP
ADD	X4	TEMP	R02	CVT1
SUB	X4	TEMP	R06	NOP
ADD	X4	TEMP	R08	NOP
SUB	X4	TEMP	R10	CVT0
		F*X2		
LDA	Y1	X2	R11	NOP
SUB	Y1	X2	R02	NOP
SUB	Y1	X2	R06	NOP
SUB	Y1	X2	R07	NOP
LDA	TEMP	X2	R13	NOP
ADD	Y1	TEMP	R02	NOP
ADD	Y1	TEMP	R04	NOP
ADD	Y1	TEMP	F07	NOP

TABLE 5 (Continued)

ADD	Y1	TEMP	R09	NOP
ADD	Y1	TEMP	R12	NOP
		G*X3		
ADD	Y1	X3	R01	NOP
SUB	Y1	X3	R03	NOP
ADD	Y1	X3	R07	NOP
ADD	Y1	X3	R08	NOP
SUB	Y1	X3	R10	NOP
ADD	Y1	X3	R12	NOP
LDA	TEMP	X3	R13	NOP
SUB	Y1	TEMP	R02	NOP
SUB	Y1	TEMP	R05	NOP
ADD	Y1	TEMP	R07	NOP
SUB	Y1	TEMP	R10	NOP
		C*X5		
ADD	Y1	X5	R01	NOP
SUB	Y1	X5	R04	NOP
SUB	Y1	X5	R06	NOP
SUB	Y1	X5	R07	NOP
ADD	Y1	X5	R12	NOP
SUB	Y1	X5	R13	NOP
LDA	TEMP	X5	R13	NOP
ADD	Y1	TEMP	R02	NOP
SUB	Y1	TEMP	R06	NOP
ADD	Y1	TEMP	R08	NOP
SUB	Y1	TEMP	R10	NOP
ADD	Y1	TEMP	R13	NOP
		DAR TO REGISTERS		
ADD	X1	DAR	R05	NOP
				NOP
				NOP
				NOP
ADD	X4	DAR	R05	NOP
				NOP
				NOP
				NOP
				NOP
LDA	X6	DAR	R05	NOP
				NOP
				NOP
				NOP
				NOP

TABLE 5 (Continued)

```
                        A*X6
ADD         Y1          X6          R05         NOP
ADD         Y1          X6          R11         NOP
LDA         TEMP        X6          R13         NOP
SUB         Y1          TEMP        R01         NOP
ADD         Y1          TEMP        R03         NOP
SUB         Y1          TEMP        R05         NOP
ADD         Y1          TEMP        R08         NOP
ADD         Y1          TEMP        R10         NOP
ADD         Y1          TEMP        R11         NOP
              SCALE OUTPUT UP BY 16
LDA         TEMP        Y1          L02         NOP
LDA         DAR         TEMP        L02         NOP
                                                  NOP
                                                  NOP
                                                  NOP
                                                  NOP
                                                  OUT1
                                                  OUT1
                                                  OUT1
LOOP THROUGH PROGRAM TWICE BEFORE UPDATING OUTPUT
SUB         CNT         KP2         R00         NOP
LDA         DAR         CNT         R00         NOP
LDA         X3          X2          R00         CNDS
LDA         X2          X1          R00         CNDS
LDA         X5          X4          R00         CNDS
LDA         CNT         KP3         R00         CNDS
                                                  EOP
                                                  NOP
                                                  NOP
                                                  NOP
                                                  END
```

TABLE 6

STEP RESPONSE FROM DIFFERENCE EQUATION AND SIMULATION

STEP INPUT:		
Count	Output of Difference	Simulation
0	.0316893439	.03125
1	.1729930493	.171875
2	.4453782974	.4453125
3	.7521160718	.75
4	.9796377451	.9765625
5	1.08623858	1.0859375
6	1.095324077	1.09375
7	1.056543501	1.0546875
8	1.012778691	1.0078125
9	.9860264349	.984375
10	.9791647132	.9765625
11	.9848496678	.984375
12	.9940987213	.9921875
13	1.001060512	1.0
14	1.003926542	1.0
15	1.003590225	1.0

values for the assembly language filter are given in Appendix D, and in Table 6. Also, see Figure 4.

$$\begin{aligned} y(k) = {} & .0316893439x(k) + 3(.0316893439)x(k-1) + \\ & 3(.0316893439)x(k-2) + .0316893439x(k-3) + \\ & 1.459029062y(k-1) - .910269y(k-2) + \\ & .1978251872y(k-3) \end{aligned} \tag{9}$$

Note that in the simulation a final steady state output of .5 is obtained. In order to prevent saturation of the device, the input is scaled down by thirty-two, and the output is scaled up by sixteen. The device will saturate if a one or greater number is ever encountered. Minus one, inclusive, is the lower bound.

As a result of initially scaling the input down by thirty-two some of the resolution is lost. A more appropriate means of scaling the input would be to start with an output from the difference equation. This would key the maximum gain. With this information, a wiser decision could be made with regards to initial scaling values.

Continued analysis of the filter could include a steady state response for a sinusoidal input; however, further analysis will not be included here.

In the next chapter, this same design example is realized using the Intel SPAS20.SFT design software. A variety of macros are available from INTEL SPAS20.software. Several are very useful, while others have very limited filter design use.

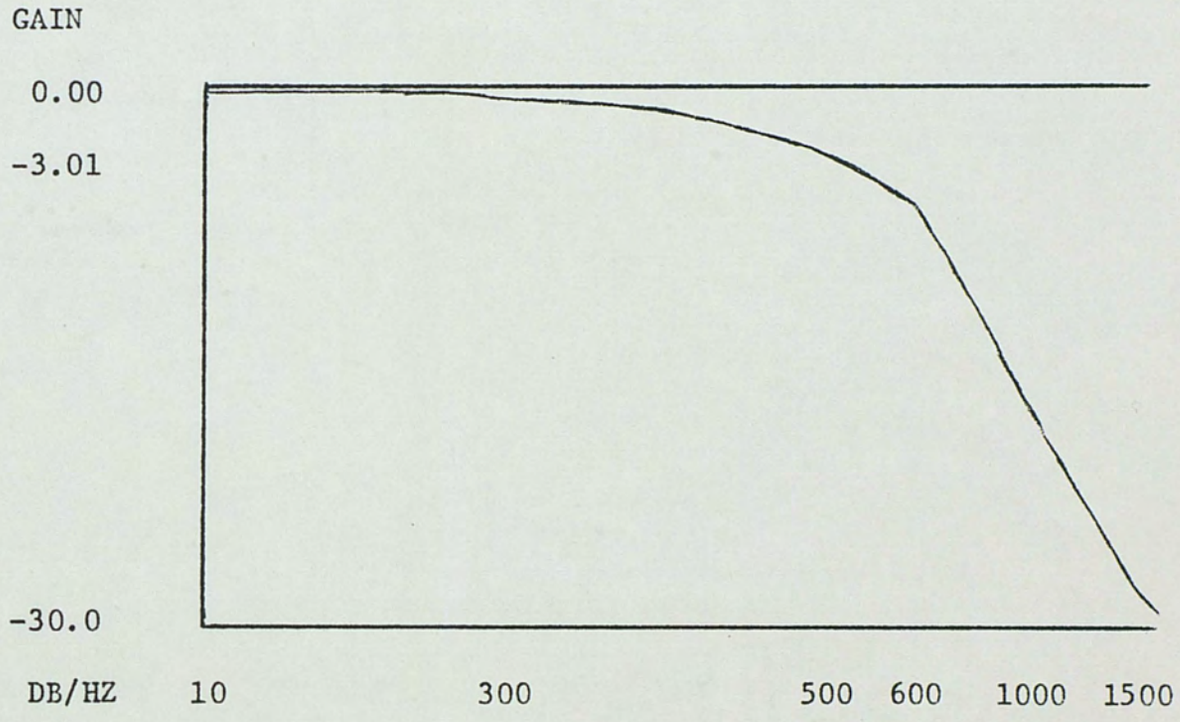

Fig. 4. A third order digital Butterworth filter frequency response.

CHAPTER II

DIGITAL FILTER DESIGN USING INTEL SOFTWARE

The purpose of this chapter is to demonstrate the use of Intel's SPAS20 digital filter computer aided design software. Included in this chapter, which will be applied to the previous example, is a discussion of the available macros for performing analog design of both Butterworth and Chebychev filters, bilinear transformation, and assembly language program generation. This chapter relies heavily on Appendix E, which is the computer design printout. A discussion of computer control, editing, assembly and simulation can be found in references 4, 5, 6 and in Appendix C.

The SPAS20.SFT design software is invoked on the Intel series 210 minicomputer by typing "SPAS20.SFT". Control and keywords can now be entered. A complete list of keywords and control commands can be found in Appendix B of reference 1.

To design Butterworth or Chebychev filters, the macro SPAS20.MC1 can be used. This macro, like all utilized macros, must be included under SPAS20.SFT control by typing "INCLUDE SPAS20.MC1".

A list of the macro and a simple procedure for utilization is printed. See Appendix E.

The third order low pass Butterworth filter poles and zeros are produced by typing ":BUTTER 3,500,1".

The colon indicates to the SPAS controller that Butter is included in a predefined macro. The 3,500,1 combination is: order of filter, cutoff frequency and the starting pole number label. The poles generated by this procedure are listed in Table 7. Notice that these poles are precisely the same as were generated in the first chapter.

TABLE 7

CONTINUOUS S-PLANE POLES

Pole 1 = -249.99995, 433.01272
Pole 2 = -500.00000, 0.0000000

The gain and step transient response is obtained by telling the computer to graph gain or step. The horizontal scale for gain can be defined by the command word FSCALE. See the example in Appendix E.

Conversion to the z-plane is performed by macro SPAS20.MC2. This predefined macro employs the constant C as two over the normalized sample frequency. Conversion of the analog poles and zeros to the z-plane is accomplished by typing ":BTP 1,100", where 1 defines pole number 1 in the continuous s-plane, and 100 defines pole number 100 in the z-plane. The poles and zeros generated in this bilinear transformation are listed in Table 8, where the poles and zeros are given in polar coordinates. Comparison of Table 8 and

Table 4 is of interest. The difference in pole/zero location is a direct result of choice of constant C used in the bilinear transform. One could use an adjusted psuedo sample period to obtain the same result.

TABLE 8

Z-PLANE POLES

Pole 100 = 0.70162724, 0.67729125
Pole 200 = .43606005, 0.0000000
Zero 100 = 1.000000, Pi
Zero 101 = 1.000000, Pi
Zero 200 = 1.000000, Pi

For comparison of gain and step response from analog to digital filters see Appendix E. The gain at five hundred hertz is listed on each of the different gain curves. Note that as a result of the choice of constant C in the bilinear transform macro, gain at five hundred hertz does not match between the analog and digital filters.

Once the poles and zeros have been located in the z-plane, the assembly language can be generated, This involves use of the SPAS20. MC3 macro. This macro utilizes the cascade canonic form as described in Figure 5.

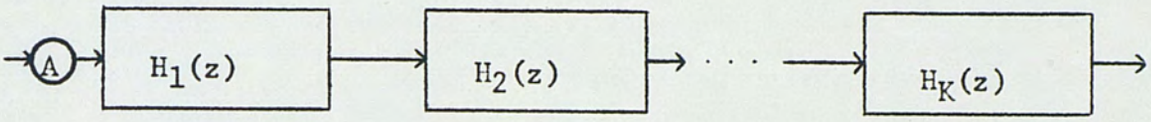

Fig. 5. Cascade realization.

where: $H(Z) = A \prod_{i=1}^{K} H_i(Z)$

In order to avoid saturation from one stage to another, the gain of each stage can be graphed or, the maximum absolute gain can be requested. Once this maximum absolute gain, magain, is determined, the stage input can be scaled accordingly. This procedure continues until the final stage.

An illustrative example of assembly language program development using this cascade form, computer design, is given in reference 1. Most of the work in assembly language development is actually a matter of finding the poles and zeros in the z-plane corresponding to the s-plane locations. Utilization of the SPAS20.MC3 software for code generation is equally time consuming as parallel canonic generation. One advantage of the software is the optimum binary expansion routine for pole, zero assembly language generation.

There is definite time savings in filter design using the available SPAS20 software. For some it may also serve as a design guide. However, this should not replace an understanding of the fundamental concepts.

The next chapter briefly reviews the results of each designed filter.

CHAPTER III

COMPARISON

The ultimate comparison of the filters should be based upon the initial filter design. If both filters were within design tolerances, then the design process which required less cost outlay should be utilized. If numerous different filters were to be designed, then becoming acquainted with the SPAS.SFT software would be worth the effort.

The analog filters designed in both Chapters I and II are identical. Correspondingly, the gain and step responses are the same.

If the digital filter in Chapter I had been designed with the same defined constant as in Chapter II, the frequency responses and step responses would also have been identical. As it is, there is a greater power loss at the cutoff frequency in the second filter.

A final comparison of the filters might include the steady state response to a sinusoidal input. This is easily determined by using the SM2920.SFT simulation software.

For the problem as originally stated, both designed filters meet the design requirements. A detailed demonstration of both fundamental concepts and design software has been presented.

CHAPTER IV

CONCLUSION

The purpose of this paper was to demonstrate the development of a digital filter with Intel's computer aided design software. The basic approach was to develop the same filter using commonly accepted analog filter designs and the Intel computer aided software. The coefficients of these filters were derived from both filter definition and existing tables. The analog filter was then transformed into a digital filter using the bilinear transform method. Step response and frequency response were then generated and compared with the computer generated step and frequency response. Assembly language for the filter was written and this was simulated with the SM2920.SFT software.

A second purpose of this paper was to collate some of the basic computer manipulation files into quick reference appendices. These include the assembly, simulation and design software. Along with these were basic instructions for access into the Series 210 microcomputer development system.

As the digital filter chips are improved and bandwidth is increased, there will be fewer and fewer analog circuits designed. The inherent reliability and replicability of digital devices makes analog to digital conversion ever more popular.

APPENDICES

APPENDIX A

ANALOG PROGRAM AND GAIN

This section contains a TI-59 program listing and frequency response for the third order Butterworth design.

TABLE 9

LIST OF TI-59 PROGRAM LISTING

LBL	RCL	05
A	02	PRT
RCL	x	RCL
05	2	09
X^2	=	PRT
z	+	RCL
RCL	RCL	04
01	07	+
x	=	2
2	X^2	5
+/-	+	=
+	RCL	STO
RCL	06	04
03	=	x
=	LOG	2
X^2	x	x
STP	1	π
06	0	=
RCL	=	STO
05	+/-	05
Y^x	RCL	GTO
3	08	00
=	=	02
+/-	STO	0
STO	09	0
07	RCL	0
RCL	04	0
05	PRT	0
x	RCL	

Butterworth: n = 3 (TI-59) Program (Analog)

APPENDIX B

COMPUTER GENERATED OBJECT FILES, RLO, LST AND HEX

The assembly language third order Butterworth filter program in Table 5 is entered onto disk by using the CREDIT control file. Enter the following command to create a new file for storage of the assembly language program:

CREDIT (6 Letter Name).(3 Letter Distinguisher)

Do not create a new file if it is already listed on the disk with CREDIT. Now, when the CURSOR appears you can type in program text. A semi colon can be used to indicate comments. Once you get this far, if you need help simply type "HELP". To exit from credit go to the control mode by touching the HOME KEY, far right of keyboard. Now type "EXIT" and the new program is copied onto your disk.

To assemble the newly generated program type in "AS2920 (6 Letter Name).(3 Letter Distinguisher)". This will produce a listing of errors and warnings and two new files, (6 Letter Name).HEX and (6 Letter Name).LST. The errors and warnings are indicated to the left of the mistake in the LST file. These can be viewed by typing "CREDIT (6 Letter Name).LST". Holding the CNTL key down while pressing N will proceed to the next page on the console. CNTL P will view the previous page. Once the corrections are noted in

the LST file go back to the original file and make corrections. Eliminate the old HEX and LST files and reassemble the corrected file. Continue until the file contains no errors or warnings.

To simulate the program, type in "SM2920.SFT". Thirty seconds later, or so, a CURSOR will appear. Now load the HEX file by typing "LOAD (6 Letter Name).HEX".

You are now ready to simulate. In order to simulate, set the following parameters to desired values:

QUALIFIER

TRACE

iN1

BREAKPOINT

RAM (0 to value listed on assembly LST file)

DAR

See Appendix D for examples of how to set these parameters.

ASSEMBLY CODE LISTING

```
;:::::::::::::::::::::::::::::::::::::::::::
;                  D*X1
;:::::::::::::::::::::::::::::::::::::::::::
LDAX1, X2, R00, IN1
ADDX1, X2, R05, IN1
ADDX1, X2, R06, IN1
SUBX1, X2, R09, NOP
SUBX1, X2, R13, NOP
LDATEMP, X2, R13, CVTS
ADDX1, TEMP, R03, NOP
ADDX1, TEMP, R08, NOP
SUBX1, TEMP, R12, CVT7
SUBX1, TEMP, R13, NOP
;:::::::::::::::::::::::::::::::::::::::::::
;                  E*X3
;:::::::::::::::::::::::::::::::::::::::::::
SUBX1, X3, R01, NOP
ADDX1, X3, R06, CVT6
ADDX1, X3, R07, NOP
SUBX1, X3, R10, NOP
LDATEMP, X3, R13, CVT5
SUBX1, TEMP, R01, NOP
ADDX1, TEMP, R04, NOP
ADDX1, TEMP, R09, CVT4
SUBX1, TEMP, R13, NOP
;:::::::::::::::::::::::::::::::::::::::::::
;                  B*X5
;:::::::::::::::::::::::::::::::::::::::::::
LDAX4, X5, R01, NOP
SUBX4, X5, R04, CVT3
SUBX4, X5, R06, NOP
SUBX4, X5, R07, NOP
ADDX4, X5, R12, CVT2
SUBX4, X5, R13, NOP
LDATEMP, X5, R13, NOP
ADDX4, TEMP, R02, CVT1
SUBX4, TEMP, R06, NOP
ADDX4, TEMP, R08, NOP
SUBX4, TEMP, R10, CVT0
ADDX4, TEMP, R13, NOP
;:::::::::::::::::::::::::::::::::::::::::::
;                  F*X2
;:::::::::::::::::::::::::::::::::::::::::::
LDAY1, X2, R11, NOP
SUBY1, X2, R02, NOP
SUBY1, X2, R06, NOP
SUBY1, X2, R07, NOP
LDATEMP, X2, R13, NOP
ADDY1, TEMP, R02, NOP
ADDY1, TEMP, R04, NOP
ADDY1, TEMP, R07, NOP
ADDY1, TEMP, R09, NOP
ADDY1, TEMP, R12, NOP
;:::::::::::::::::::::::::::::::::::::::::::
;                  G*X3
;:::::::::::::::::::::::::::::::::::::::::::
```

ASSEMBLY CODE LISTING CONTINUED

```
ADDY1, X3, R01, NOP
SUBY1, X3, R03, NOP
ADDY1, X3, R07, NOP
ADDY1, X3, R08, NOP
SUBY1, X3, R10, NOP
ADDY1, X3, R12, NOP
LDATEMP, X3, R13, NOP
SUBY1, TEMP, R02, NOP
SUBY1, TEMP, R05, NOP
ADDY1, TEMP, R07, NOP
SUBY1, TEMP, R10, NOP
;:::::::::::::::::::::::::::::::::::::::::::::
;                C*X5
;:::::::::::::::::::::::::::::::::::::::::::::
ADDY1, X5, R01, NOP
SUBY1, X5, R04, NOP
SUBY1, X5, R06, NOP
SUBY1, X5, R07, NOP
ADDY1, X5, R12, NOP
SUBY1, X5, R13, NOP
LDATEMP, X5, R13, NOP
ADDY1, TEMP, R02, NOP
SUBY1,       TEMP, R06, NOP
ADDY1, TEMP, R08, NOP
SUBY1, TEMP, R10, NOP
ADDY1, TEMP, R13, NOP
;:::::::::::::::::::::::::::::::::::::::::::::
;                DAR TO REGISTERS
;:::::::::::::::::::::::::::::::::::::::::::::
ADDX1, DAR, R05, NOP
NOP
NOP
NOP
NOP
ADDX4, DAR, R05, NOP
NOP
NOP
NOP
NOP
LDAX6, DAR, R05, NOP
NOP
NOP
NOP
NOP
;:::::::::::::::::::::::::::::::::::::::::::::
;                A*X6
;:::::::::::::::::::::::::::::::::::::::::::::
ADDY1, X6, R05, NOP
ADDY1, X6, R11, NOP
LDATEMP, X6, R13, NOP
SUBY1, TEMP, R01, NOP
ADDY1, TEMP, R03, NOP
SUBY1, TEMP, R05, NOP
ADDY1, TEMP, R08, NOP
ADDY1, TEMP, R10, NOP
ADDY1, TEMP, R11, NOP
```

ASSEMBLY CODE LISTING CONTINUED

```
;:::::::::::::::::::::::::::::::::::::::::::::
;                SCALE OUTPUT BY 16
;:::::::::::::::::::::::::::::::::::::::::::::
LDATEMP,Y1,L02,NOP
LDADAR,TEMP,L02,NOP
NOP
NOP
NOP
NOP
OUT1
OUT1
OUT1
;:::::::::::::::::::::::::::::::::::::::::::::
;    LOOP THRU PROGRAM TWICE BEFORE UPDATING
;:::::::::::::::::::::::::::::::::::::::::::::
SUBCNT,KP2,R00,NOP
LDADAR,CNT,R00,NOP
LDAX3,X2,R00,CNDS
LDAX2,X1,R00,CNDS
LDAX5,X4,R00,CNDS
LDACNT,KP3,R00,CNDS
NOP
EOP
NOP
NOP
NOP
END
```

```
ISIS-II 2920 ASSEMBLER V1.0                                          PAGE   1

ASSEMBLER INVOKED BY: AS2920 PAPA1.RLO


  LINE  LOC OBJECT SOURCE STATEMENT

     1            ;::::::::::::::::::::::::::::::::::::::::
     2            ;              D*X1
     3            ;::::::::::::::::::::::::::::::::::::::::
     4     0 1008EF LDA X1, X2, R00,     IN1
     5     1 10088C ADD X1, X2, R05,     IN1
     6     2 1008AC ADD X1, X2, R06,     IN1
     7     3 40080B SUB X1, X2, R09,     NOP
     8     4 40088B SUB X1, X2, R13,     NOP
     9     5 64088F LDA TEMP,   X2, R13,     CVT5
    10     6 42004C ADD X1, TEMP,   R03,     NOP
    11     7 4200EC ADD X1, TEMP,   R08,     NOP
    12     8 73006B SUB X1, TEMP,   R12,     CVT7
    13     9 42008B SUB X1, TEMP,   R13,     NOP
    14            ;::::::::::::::::::::::::::::::::::::::::
    15            ;              E*X3
    16            ;::::::::::::::::::::::::::::::::::::::::
    17    10 42080A SUB X1, X3, R01,     NOP
    18    11 6308AC ADD X1, X3, R06,     CVT6
    19    12 4208CC ADD X1, X3, R07,     NOP
    20    13 42082B SUB X1, X3, R10,     NOP
    21    14 57088F LDA TEMP,   X3, R13,     CVT5
    22    15 42000A SUB X1, TEMP,   R01,     NOP
    23    16 4200EC ADD X1, TEMP,   R04,     NOP
    24    17 43000D ADD X1, TEMP,   R09,     CVT4
    25    18 42008B SUB X1, TEMP,   R13,     NOP
    26            ;::::::::::::::::::::::::::::::::::::::::
    27            ;              B*X5
    28            ;::::::::::::::::::::::::::::::::::::::::
    29    19 48180E LDA X4, X5, R01,     NOP
    30    20 39186A SUB X4, X5, R04,     CVT3
    31    21 4818AA SUB X4, X5, R06,     NOP
    32    22 4818CA SUB X4, X5, R07,     NOP
    33    23 29186D ADD X4, X5, R12,     CVT2
    34    24 48188B SUB X4, X5, R13,     NOP
    35    25 4C088F LDA TEMP,   X5, R13,     NOP
    36    26 13102C ADD X4, TEMP,   R02,     CVT1
    37    27 4210AA SUB X4, TEMP,   R06,     NOP
    38    28 4210EC ADD X4, TEMP,   R08,     NOP
    39    29 03102B SUB X4, TEMP,   R10,     CVT0
    40    30 42108D ADD X4, TEMP,   R13,     NOP
    41            ;::::::::::::::::::::::::::::::::::::::::
    42            ;              F*X2
    43            ;::::::::::::::::::::::::::::::::::::::::
    44    31 44184F LDA Y1, X2, R11,     NOP
    45    32 44182A SUB Y1, X2, R02,     NOP
    46    33 4418AA SUB Y1, X2, R06,     NOP
    47    34 4418CA SUB Y1, X2, R07,     NOP
    48    35 44088F LDA TEMP,   X2, R13,     NOP
    49    36 46102C ADD Y1, TEMP,   R02,     NOP
    50    37 46106C ADD Y1, TEMP,   R04,     NOP
    51    38 4610CC ADD Y1, TEMP,   R07,     NOP
    52    39 46100D ADD Y1, TEMP,   R09,     NOP
```

```
ISIS-II 2920 ASSEMBLER V1.0                                    PAGE   2

 LINE  LOC OBJECT SOURCE STATEMENT

   53   40 46106D ADD Y1, TEMP,    R12,     NOP
   54             ;::::::::::::::::::::::::::::::::::::::::::::
   55             ;                  G*X3
   56             ;::::::::::::::::::::::::::::::::::::::::::::
   57   41 46180C ADD Y1, X3, R01,     NOP
   58   42 46184A SUB Y1, X3, R03,     NOP
   59   43 4618CC ADD Y1, X3, R07,     NOP
   60   44 4618EC ADD Y1, X3, R08,     NOP
   61   45 46182B SUB Y1, X3, R10,     NOP
   62   46 46186D ADD Y1, X3, R12,     NOP
   63   47 46088F LDA TEMP,    X3, R13,     NOP
   64   48 46102A SUB Y1, TEMP,    R02,     NOP
   65   49 46108A SUB Y1, TEMP,    R05,     NOP
   66   50 4610CC ADD Y1, TEMP,    R07,     NOP
   67   51 46102B SUB Y1, TEMP,    R10,     NOP
   68             ;::::::::::::::::::::::::::::::::::::::::::::
   69             ;                  C*X5
   70             ;::::::::::::::::::::::::::::::::::::::::::::
   71   52 4C180C ADD Y1, X5, R01,     NOP
   72   53 4C186A SUB Y1, X5, R04,     NOP
   73   54 4C18AA SUB Y1, X5, R06,     NOP
   74   55 4C18CA SUB Y1, X5, R07,     NOP
   75   56 4C186D ADD Y1, X5, R12,     NOP
   76   57 4C188B SUB Y1, X5, R13,     NOP
   77   58 4C088F LDA TEMP,    X5, R13,     NOP
   78   59 46102C ADD Y1, TEMP,    R02,     NOP
   79   60 4610AA SUB Y1,      TEMP,    R06,     NOP
   80   61 4610EC ADD Y1, TEMP,    R08,     NOP
   81   62 46102B SUB Y1, TEMP,    R10,     NOP
   82   63 46108D ADD Y1, TEMP,    R13,     NOP
   83             ;::::::::::::::::::::::::::::::::::::::::::::
   84             ;                DAR TO REGISTERS
   85             ;::::::::::::::::::::::::::::::::::::::::::::
   86   64 40228C ADD X1, DAR,     R05,     NOP
   87   65 4000EF                  NOP
   88   66 4000EF                  NOP
   89   67 4000EF                  NOP
   90   68 4000EF                  NOP
   91   69 40328C ADD X4, DAR,     R05,     NOP
   92   70 4000EF                  NOP
   93   71 4000EF                  NOP
   94   72 4000EF                  NOP
   95   73 4000EF                  NOP
   96   74 44329E LDA X6, DAR,     R05,     NOP
   97   75 4000EF                  NOP
   98   76 4000EF                  NOP
   99   77 4000EF                  NOP
  100   78 4000EF                  NOP
  101             ;::::::::::::::::::::::::::::::::::::::::::::
  102             ;                A*X6
  103             ;::::::::::::::::::::::::::::::::::::::::::::
  104   79 4E188C ADD Y1, X6, R05,     NOP
  105   80 4E184D ADD Y1, X6, R11,     NOP
  106   81 4E088F LDA TEMP,    X6, R13,     NOP
```

```
ISIS-II 2920 ASSEMBLER V1.0                                          PAGE   3


  LINE  LOC OBJECT SOURCE STATEMENT

   107   82 46100A SUB Y1, TEMP,    R01,     NOP
   108   83 46104C ADD Y1, TEMP,    R03,     NOP
   109   84 46108A SUB Y1, TEMP,    R05,     NOP
   110   85 4610EC ADD Y1, TEMP,    R08,     NOP
   111   86 46102D ADD Y1, TEMP,    R10,     NOP
   112   87 46104D ADD Y1, TEMP,    R11,     NOP
   113              ;:::::::::::::::::::::::::::::::::::::::::::::
   114              ;                SCALE OUTPUT BY 16
   115              ;:::::::::::::::::::::::::::::::::::::::::::::
   116   88 4E00AF LDA TEMP,    Y1, L02,     NOP
   117   89 4244AF LDA DAR,     TEMP,    L02,     NOP
   118   90 4000EF                NOP
   119   91 4000EF                NOP
   120   92 4000EF                NOP
   121   93 4000EF                NOP
   122   94 9000EF                OUT1
   123   95 9000EF                OUT1
   124   96 9000EF                OUT1
   125              ;:::::::::::::::::::::::::::::::::::::::::::::
   126              ;   LOOP THRU PROGRAM TWICE BEFORE UPDATING
   127              ;:::::::::::::::::::::::::::::::::::::::::::::
   128   97 42C2EB SUB CNT,     KP2,     R00,     NOP
   129   98 4064EF LDA DAR,     CNT,     R00,     NOP
   130   99 7408FF LDA X3, X2,  R00,     CNDS
   131  100 7000FF LDA X2, X1,  R00,     CNDS
   132  101 7810FF LDA X5, X4,  R00,     CNDS
   133  102 72CAEF LDA CNT,     KP3,     R00,     CNDS
   134  103 4000EF                NOP
   135  104 5000EF                EOP
   136  105 4000EF                NOP
   137  106 4000EF                NOP
   138  107 4000EF                NOP
   139                           END

SYMBOL:                          VALUE:

X1                                  0
X2                                  1
TEMP                                2
X3                                  3
X4                                  4
X5                                  5
Y1                                  6
X6                                  7
CNT                                 8


ASSEMBLY COMPLETE
ERRORS    =    0
WARNINGS  =    0
RAMSIZE   =    9
ROMSIZE   =  108
:18000000F1F0F0F8FEFFF1F0F0F8F8FCF1F0F0F8FAFCF4F0F0F8F0FBEF
```

ASSEMBLED HEX CODE

```
:18001800F4F0F0F8F8FBF6F4F0F8F8FFF4F2F0F0F4FCF4F2F0F0FEFCD2
:18003000F7F3F0F0F6FBF4F2F0F0F8FBF4F2F0F8F0FAF6F3F0F8FAFCC5
:18004800F4F2F0F8FCFCF4F2F0F8F2FBF5F7F0F8F8FFF4F2F0F0F0FAA4
:18006000F4F2F0F0F6FCF4F3F0F0F0FDF4F2F0F0F8FBF4F8F1F8F0FEA0
:18007800F3F9F1F8F6FAF4F8F1F8FAFAF4F8F1F8FCFAF2F9F1F8F6FD50
:18009000F4F8F1F8F8FBF4FCF0F8F8FFF1F3F1F0F2FCF4F2F1F0FAFA53
:1800A800F4F2F1F0FEFCF0F3F1F0F2FBF4F2F1F0F8FDF4F4F1F8F4FF4E
:1800C000F4F4F1F8F2FAF4F4F1F8FAFAF4F4F1F8FCFAF4F4F0F8F8FF18
:1800D800F4F6F1F0F2FCF4F6F1F0F6FCF4F6F1F0FCFCF4F6F1F0F0FD1F
:1800F000F4F6F1F0F6FDF4F6F1F8F0FCF4F6F1F8F4FAF4F6F1F8FCFCEF
:18010800F4F6F1F8FEFCF4F6F1F8F2FBF4F6F1F8F6FDF4F6F0F8F8FFC3
:18012000F4F6F1F0F2FAF4F6F1F0F8FAF4F6F1F0FCFCF4F6F1F0F2FBD8
:18013800F4FCF1F8F0FCF4FCF1F8F6FAF4FCF1F8FAFAF4FCF1F8FCFA85
:18015000F4FCF1F8F6FDF4FCF1F8F8FBF4FCF0F8F8FFF4F6F1F0F2FC77
:18016800F4F6F1F0FAFAF4F6F1F0FEFCF4F6F1F0F2FBF4F6F1F0F8FD83
:18018000F4F0F2F2F8FCF4F0F0F0FEFFF4F0F0F0FEFFF4F0F0F0FEFF68
:18019800F4F0F0F0FEFFF4F0F3F2F8FCF4F0F0F0FEFFF4F0F0F0FEFF4F
:1801B000F4F0F0F0FEFFF4F0F0F0FEFFF4F4F3F2F9FEF4F0F0F0FEFF30
:1801C800F4F0F0F0FEFFF4F0F0F0FEFFF4F0F0F0FEFFF4FEF1F8F8FC0D
:1801E000F4FEF1F8F4FDF4FEF0F8F8FFF4F6F1F0F0FAF4F6F1F0F4FCFA
:1801F800F4F6F1F0F8FAF4F6F1F0FEFCF4F6F1F0F2FDF4F6F1F0F4FDF7
:18021000F4FEF0F0FAFFF4F2F4F4FAFFF4F0F0F0FEFFF4F0F0F0FEFFC2
:18022800F4F0F0F0FEFFF4F0F0F0FEFFF9F0F0F0FEFFF9F0F0F0FEFFB0
:18024000F9F0F0F0FEFFF4F2FCF2FEFBF4F0F6F4FEFFF7F4F0F8FFFF77
:18025800F7F0F0F0FFFFF7F8F1F0FFFFF7F2FCFAFEFFF4F0F0F0FEFF5E
:18027000F5F0F0F0FEFFF4F0F0F0FEFFF4F0F0F0FEFFF4F0F0F0FEFF71
:00000001FF
```

APPENDIX C

CREDIT, ASSEMBLY AND SIMULATION

This appendix will give a brief overview of the Credit, Assembly and Simulation files and software. These files are very useful in the actual filter design process. Credit is equipped with editing capabilities. Credit can be used to open new files and review old files. After the assembly language program is written, the program can be entered into a new file and prepared for assembly using Credit.

The assembly program is fitted with warning and error messages to aid in debugging programs. It prepares two new files. The LST file identifies the ROM locations of the original file and all warning and error messages. The HEX file can be loaded into the simulation program for doing software simulation.

The simulation software is designed for maximum flexibility in monitoring internal calculations and output. The simulation software is invoked by typing "SM2920.SFT". This loads the control software and is now prepared for the loading of the HEX file. To load the HEX file simply type "LOAD (six letter name).HEX".

APPENDIX D

SIMULATION OF FILTER: STEP RESPONSE

This section contains a simulation run of the program in Table 5. The input was .5. Sample frequency is 4342 Hertz, due to limitation in the hardware.

SIMULATION OF FILTER: STEP RESPONSE

```
TPROG=2*4*192/6670000
*IN1
IN1 =  .99998 =  0.99997998
*DAR=0
*RAM 0 TO 8 =0
*RAM 0 TO 8
RAM 00 =  0.00000000
RAM 01 =  0.00000000
RAM 02 =  0.00000000
RAM 03 =  0.00000000
RAM 04 =  0.00000000
RAM 05 =  0.00000000
RAM 06 =  0.00000000
RAM 07 =  0.00000000
RAM 08 =  0.00000000
*OUT1
OUT1 =  0.00000000
*TRACE =T,OUT1
*Q=PC=95
*B
BREAKPOINT = NEVER
*S FROM 0
   T                  OUT1
SIMULATION BEGUN
    0.00020257         0.01562500
    0.00043285         0.08593750
    0.00066314         0.08593750
    0.00089342         0.22265625
    0.00112370         0.22265625
    0.00135399         0.37500000
    0.00158427         0.37500000
    0.00181456         0.48828125
    0.00204484         0.48828125
    0.00227513         0.54296875
    0.00250541         0.54296875
    0.00273570         0.54687500
    0.00296598         0.54687500
    0.00319627         0.52734375
    0.00342655         0.52734375
    0.00365684         0.50390625
    0.00388712         0.50390625
    0.00411741         0.49218750
    0.00434769         0.49218750
    0.00457798         0.48828125
    0.00480826         0.48828125
    0.00503855         0.49218750
    0.00526883         0.49218750
    0.00549912         0.49609375
    0.00572940         0.49609375
    0.00595969         0.50000000
    0.00618997         0.50000000
    0.00642026         0.50000000
    0.00665054         0.50000000
    0.00688083         0.50000000
    0.00711111         0.50000000
    0.00734140         0.50000000
    0.00757168         0.50000000
```

APPENDIX E

PRINTOUT OF FILTER DESIGN SESSION

This appendix contains generation of the analog Butterworth filter, bilinear transform and frequency response and step response. This software is invoked by typing "SM2920.SFT". Once this is done, we type in the separate macros by typing "INCLUDE SPAS20.MC1" for the Butterworth filter generation and "INCLUDE SPAS20.MC2" for bilinear transform.

```
*INCLUDE SPAS20.MC1
*
*;SPAS20.MC1        CONTAINS
*;                BUTTER   : GENERATE P/Z FOR BUTTERWORTH FILTER
*;                CHEB     : GENERATE P/Z FOR CHEBYSHEV   FILTER
*
*;*************************************************************************
*
*DEFINE MACRO BUTTER ; V 2.0
.*; THIS IS A BUTTERWORTH FILTER GENERATOR FOR SPAS20;
.*;
.*; CALLING SEQUENCE :BUTTER ORDER, FCO, LABEL WHERE
.*;    :BUTTER CALLS THE MACRO,
.*;     ORDER IS THE ORDER OF THE FILTER
.*;     FCO IS THE CUT-OFF FREQUENCY IN HZ
.*;     LABEL IS STARTING POINT FOR POLE NUMBERING.
.*;
.*;        EXAMPLE      :BUTTER 6,500,234
.*;                      THIS WILL GENERATE A BUTTERWORTH FILTER
.*;                      OF ORDER 6, CUTOFF=500 HZ , PRODUCING
.*;                      3 COMPLEX POLES LABELED 234,235,236
.*;
.* DEFINE .?BUTSTART = ( HPI ) + ( HPI/%0 )     ;** BEGIN THE
.* DEFINE .?BUTDELTA = ( PI/%0 )                ;** BUTTERWORTH;
.* DEFINE .?BUTINDEX = 0                        ;**   INITIALIZE
.* DEFINE .?BUTANGLE = 0                        ;**   VARIABLES;
.* REPEAT                                       ;** BEGIN LOOP,
.*   .?BUTINDEX = .?BUTINDEX + 1                ;**  CORRECT FOR
.*   .?BUTANGLE = .?BUTSTART - .?BUTINDEX * .?BUTDELTA ;* SMALL
.*   IF .?BUTANGLE < .?BUTDELTA/4 THEN          ;**  ANGLE ERROR;
.*        .?BUTANGLE=0
.*   END                                        ;** NOW CREATE
.*   DEF POL(.?BUTINDEX+%2-1) = -%1*COS(.?BUTANGLE),&     THE
.**                  %1*SIN(.?BUTANGLE)         ;**  NEXT POLE;
.*   WHILE .?BUTINDEX + 1 <= ( %0 + 1 ) / 2     ;**  CONTINUE ?
.* END                                          ;** END OF LOOP;
.* REM .?BUTANGLE                               ;**  REMOVE ALL
.* REM .?BUTINDEX                               ;**  VARIABLES
.* REM .?BUTDELTA                               ;** OF THIS MACRO;
.* REM .?BUTSTART                               ;**DISPLAY POLES;
.*PZ                                            ;**END BUTTERWORTH
.*EM  ;
*
+
*;*************************************************************************
*DEF MAC CHEB  ; V 2.0
.*;      A CHEBYSHEV FILTER GENERATOR FOR SPAS20
.*;
.*; CALLING SEQUENCE  :CHEB ORDER, FCO, LABEL, R.F.
.*; WHERE ORDER IS THE ORDER OF THE FILTER
.*;        FCO IS THE CUTOFF FREQUENCY IN HZ
.*;        LABEL IS THE STARTING POINT FOR POLE NUMBERING
.*;        R.F.  IS THE DESIRED (OR ALLOWABLE) RIPPLE FACTOR IN DB
.*
```

```
.*;         EXAMPLE    :CHEB 6,500,23,0.12
.*;                      THIS WILL GENERATE A CHEBYSHEV FILTER OF ORDER
.*;                      6, CUTOFF=500, AND A PEAK-TO-PEAK RIPPLE OF 0.12
.*;                      DB, PRODUCING 3 COMPLEX POLES LABELED 23,24,25
.*;         CALLS SUB-MACRO TEMCHB
.*;
.*DEF .?CHEBYRIP=10**(ABS(%3)/10)-1                          ;* BEGIN THE
.*DEF .?SINHP=1                                              ;* CHEBYSHEV
.*DEF .?COSHP=1                                              ;* BY SETTING
.* IF .?CHEBYRIP <> 0 THEN                                   ;* DEFAULT VALUES,
.*     :TEMCHB .?SINHP,.?COSHP,.?CHEBYRIP,%0                 ;** OR USE THE
.* END                                                       ;** SUB-MACRO
.*REM .?CHEBYRIP                                            ;** TO GENERATE
.* DEFINE .?BUTSTART = ( HPI ) + ( HPI/%0 )                  ;** THE VARIABLES.
.* DEFINE .?BUTDELTA = ( PI/%0 )
.* DEFINE .?BUTINDEX = 0                                     ;* A MODIFIED
.* DEFINE .?BUTANGLE=0                                       ;* BUTTERWORTH
.* REPEAT                                                    ;* MODULE IS
.*    .?BUTINDEX = .?BUTINDEX + 1                           ;* INCORPORATED
.*    .?BUTANGLE=.?BUTSTART - .?BUTINDEX*.?BUTDELTA         ;* TO GENERATE
.*      IF .?BUTANGLE < .?BUTDELTA/4 THEN                   ;* THE APPROPRIATE
.*          .?BUTANGLE=0                                    ;* PATTERN OF
.*      END                                                 ;* POLES FOR THE
.*    DEF POL(.?BUTINDEX+%2-1)=                  &          ;* FILTER.  (THE &S
.**           -%1*.?SINHP*COS(.?BUTANGLE),      &           ;* ALLOW GREATER
.**                 %1*.?COSHP*SIN(.?BUTANGLE)             ;* READABILITY OF
.*    WHILE .?BUTINDEX + 1 <= ( %0 + 1 ) / 2                ;* THE FORMULA.)
.* END
.* REM .?BUTINDEX                                           ;** REMOVE THE
.* REM .?BUTDELTA                                           ;** VARIABLES
.* REM .?BUTSTART                                           ;** INTRODUCED
.* REM .?BUTANGLE                                           ;** IN THIS MACRO,
.* REM .?SINHP                                              ;** TO SAVE SPACE.
.* REM .?COSHP
.*PZ
.*EM                                  ;******* END OF CHEBYSHEV MACRO ********
*
*;   TEMCHB      GET VARIABLES FOR CHEBYSHEV FILTER
*DEF MAC TEMCHB
.*;  CALLING SEQUENCE                                       ## THIS IS THE
.*;       :TEMCHB    SINHP,COSHP,E**2,N                     ## SUB-MACRO.
.*DEF .?INVSXTEMP=(1/SQR(%2))+(SQR((1/%2)+1))
.*DEF .?INVSXTEMPP=.?INVSXTEMP **(1/%3)
.*DEF .?INVSXTEMPN=.?INVSXTEMP **(-1/%3)
.*%0=(.?INVSXTEMPP - .?INVSXTEMPN)/2
.*%1=(.?INVSXTEMPP + .?INVSXTEMPN)/2
.*REM .?INVSXTEMP
.*REM .?INVSXTEMPP
.*REM .?INVSXTEMPN                                          ;## END OF TEMCHB
.*EM
*
*;*************************************************************************
*
```

```
*:BUTTER 3,500,1
.*; THIS IS A BUTTERWORTH FILTER GENERATOR FOR SPAS20;
.*;
.*; CALLING SEQUENCE :BUTTER ORDER, FCO, LABEL WHERE
.*;    :BUTTER CALLS THE MACRO,
.*;     ORDER IS THE ORDER OF THE FILTER
.*;     FCO IS THE CUT-OFF FREQUENCY IN HZ
.*;     LABEL IS STARTING POINT FOR POLE NUMBERING.
.*;
.*;        EXAMPLE      :BUTTER 6,500,234
.*;                      THIS WILL GENERATE A BUTTERWORTH FILTER
.*;                      OF ORDER 6, CUTOFF=500 HZ , PRODUCING
.*;                      3 COMPLEX POLES LABELED 234,235,236
.*;
.* DEFINE .?BUTSTART = ( HPI ) + ( HPI/3 )     ;** BEGIN THE
.* DEFINE .?BUTDELTA = ( PI/3 )                ;** BUTTERWORTH;
.* DEFINE .?BUTINDEX = 0                        ;**   INITIALIZE
.* DEFINE .?BUTANGLE = 0                        ;**   VARIABLES;
.* REPEAT                                       ;** BEGIN LOOP;
..*   .?BUTINDEX = .?BUTINDEX + 1               ;**  CORRECT FOR
..*   .?BUTANGLE = .?BUTSTART - .?BUTINDEX * .?BUTDELTA ;* SMALL
..*   IF .?BUTANGLE < .?BUTDELTA/4 THEN         ;**  ANGLE ERROR;
...*        .?BUTANGLE=0
...*   END                                      ;** NOW CREATE
..*   DEF POL(.?BUTINDEX+1-1) = -500*COS(.?BUTANGLE),&     THE
..**                    500*SIN(.?BUTANGLE)     ;**  NEXT POLE;
..*   WHILE .?BUTINDEX + 1 <= ( 3 + 1 ) / 2     ;**  CONTINUE ?
..* END                                        ;** END OF LOOP;
.* REM .?BUTANGLE                             ;**  REMOVE ALL
.* REM .?BUTINDEX                             ;**  VARIABLES
.* REM .?BUTDELTA                             ;** OF THIS MACRO;
.* REM .?BUTSTART                             ;**DISPLAY POLES;
.*PZ                                          ;**END BUTTERWORTH
.*EM  ;
POLE 1 = -249.99995,433.01272,CONTINUOUS
POLE 2 = -500.00000,0.00000000,CONTINUOUS;  REAL
```

```
*
*
*
*
*
*
*
*
*FSCALE=10,100,200,300,400,500,600,1000,1500
*
*
*                                                GRAPH GAIN

GAIN        !...^...!...^....!...^...!...^....!...^...!...^....!.......!.......!
  0.0       ////////////////////////////______...
 -1.4                                        ''--..
 -2.7                                            '--.
 -4.1                                               '--.
 -5.5                                                  '-.
 -6.8                                                    .
 -8.2                                                    .
 -9.5                                                    .
-10.9                                                    '
-12.3                                                     '
-13.6                                                      -
-15.0                                                      -
-16.4                                                       .
-17.7                                                       .
-19.1                                                       .
-20.5                                                       -
-21.8                                                       -
-23.2                                                       -
-24.5                                                       '
-25.9                                                       '
-27.3                                                       '.
-28.6                                                       .
    DB\HZ   !...^...!...^....!...^...!...^....!...^...!...^....!.......!.......!
*           10 30 100 140  200 240 300 340  400 450 500 540  600    1000     1500
*
*
*
*
*
*GAIN(500)
GAIN(500.00000) = -3.0102994
```

```
TS=1/4000
TS = 2.4999995/10**4
*XSIZE
XSIZE = 79.000000
*
*
*
*
*
*
*
*
*
*
*                                                    GRAPH STEP

STEP       !.....^.....^.....^.....^.....^.....^.....^.....!
1.49            /-
1.42           -  /.....................................................
1.35                ////
1.28
1.21          .
1.14
1.07
0.99
0.92
0.85
0.78          .
0.71
0.64
0.57
0.50
0.43
0.36
0.28         -
0.21
0.14
0.07
0.00          .
  AMP\SEC  !.....^.....^.....^.....^.....^.....^.....^.....!
           0.000   0.003   0.005   0.007 0.0087 0.01   0.012   0.014        0.017
```

```
P
POLE 1 = -249.99995,433.01272,CONTINUOUS
POLE 2 = -500.00000,0.00000000,CONTINUOUS;  REAL
*Z
*INCLUDE SPAS20.MC2
*
*;SPAS20.MC2 CONTAINS
*;                  BTP  :  BILINEAR TRANSFORM A POLE
*;                  BTZ  :  BILINEAR TRANSFORM A ZERO
*;
*;*************************************************************************
*DEF MACRO BTP   ;  V2.0
.*          ;THIS MACRO GENERATES A BILINEAR TRANSFORM OF A GIVEN POLE
.*          ;CALLING SEQUENCE     :BTP  POLE # IN S, POLE # IN Z
.*          ;EXAMPLE              :BTP  3,90
.*          ;                     THIS WILL TAKE A POLE IN THE S PLANE,(P 3)SPECIFIE
.*          ;                     IN HZ, AND PRODUCE A POLE IN THE Z PLANE (P 90)
.*          ;                     AND 1 OR 2 EXTRA ZEROES (Z 90 / Z 91)
.*          ;                     DEPENDING ON WHETHER P 3 IS A REAL POLE OR COMPLEX
.*          ;  CALLS SUB-MACRO BLTCOM, CKREAL
.*:BLTCOM %0, %1, POLE, ZERO
.*EM
*
*
*
*DEF MACRO BTZ   ;  V 2.0
.*          ;THIS MACRO GENERATES A BILINEAR TRANSFORM OF A GIVEN ZERO
.*          ;CALLING SEQUENCE       :BTZ  ZERO # IN S, ZERO # IN Z
.*          ;EXAMPLE                :BTZ  3,90
.*          ;                       TAKES A ZERO IN THE S PLANE,(Z 3)SPECIFIED IN H
.*          ;                       AND PRODUCE A ZERO IN THE Z PLANE (Z 90)
.*          ;                       AND 1 OR 2 EXTRA POLES (P 90 / P 91)
.*          ;                       DEPENDING ON WHETHER Z 3 IS REAL OR COMPLEX
.*          ;      CALLS SUB-MACRO BLTCOM,CKREAL
.*:BLTCOM %0, %1, ZERO, POLE
.*EM
*
*
*
*DEF MACRO CKREAL   ;  V 2.0
.*;  SUB MACRO USED IN ANOTHER SUB-MACRO BLTCOM TO CHECK FOR REAL POLE/ZERO AND A
.*;             ANGLE ETC
.*IF IMAG(%2 %0)=0 THEN
.*       REM %3 %1+1          ;REMOVE EXTRA PZ
.*       %5=SQR(%5)           ;ADJUST SCALE FACTOR
.*       IF ABS(TPI*REAL(%2 %0))<%4 THEN    ;ADJUST ANGLE
.*                  MOV  %2 %1 TO RAD(%2 %1),0
.*       ELSE
.*                  MOV  %2 %1 TO RAD(%2 %1),PI
.*       END
.*END
.*EM
*;
*;
*
*DEF MACRO BLTCOM   ;  V 2.0
```

```
.*; THIS MACRO IS CALLED FROM BTP/BTZ, WHICH SUPPLY POLE/ZERO
.*; LABELS ,AND THE PROPER 'POLE'/'ZERO'
.*
.*; THE TRANSFORM EQUATION HERE IS
.*;
.*;            S  = (C) * (1-Z**-1) / (1 + Z**-1)
.*;
.*DEF .?QC = 2/TS                          ;THIS IS THE CONSTANT C USED IN THIS MA
.*DEF .?QA = - REAL (%2 %0)*TPI
.*DEF .?QB = IMAG (%2 %0)*TPI
.*DEF .?QA0 = (.?QC + .?QA)**2 + .?QB **2
.*DEF .?QA1 = 2*(.?QA**2 + .?QB**2 - .?QC**2)
.*DEF .?QA2 = (.?QC-.?QA)**2 + .?QB**2
.*DEF .?QRADIUS = SQR (.?QA2/.?QA0)
.*DEF .?QANGLE = -.?QA1/ (.?QA0*2*.?QRADIUS)
.*IF .?QANGLE>1
.*.?QANGLE=1
.*ORIF .?QANGLE<-1
.*.?QANGLE=-1
.*END
.*.?QANGLE = ACOS (.?QANGLE)
.*DEF %2 %1 = .?QRADIUS, .?QANGLE, Z
.*DEF %3 %1 = 1, PI, Z
.*DEF %3 (%1+1) = 1, PI, Z
.*;
.*;ADJUST FOR REAL IF NECESSARY
.*:CKREAL %0,%1,%2,%3,.?QC,.?QA0
.*;
.*WRITE 'SCALE FACTOR = ', 1/.?QA0
.*;
.*REM .?QC
.*REM .?QA
.*REM .?QB
.*REM .?QA0
.*REM .?QA1
.*REM .?QA2
.*REM .?QRADIUS
.*REM .?QANGLE
.*EM
*;
*;*************************************************************************************
*
```

```
:BTP 1,100
.*          ;THIS MACRO GENERATES A BILINEAR TRANSFORM OF A GIVEN POLE
.*          ;CALLING SEQUENCE    :BTP  POLE # IN S, POLE # IN Z
.*          ;EXAMPLE             :BTP  3,90
.*          ;                    THIS WILL TAKE A POLE IN THE S PLANE,(P 3)SPECIFIE
.*          ;                    IN HZ, AND PRODUCE A POLE IN THE Z PLANE (P 90)
.*          ;                    AND 1 OR 2 EXTRA ZEROES (Z 90 / Z 91)
.*          ;                    DEPENDING ON WHETHER P 3 IS A REAL POLE OR COMPLEX
.*          ; CALLS SUB-MACRO BLTCOM, CKREAL
.*:BLTCOM 1, 100, POLE, ZERO
..*; THIS MACRO IS CALLED FROM BTP/BTZ, WHICH SUPPLY POLE/ZERO
..*; LABELS ,AND THE PROPER 'POLE'/'ZERO'
..*
..*; THE TRANSFORM EQUATION HERE IS
..*;
..*;            S  = (C) * (1-Z**-1) / (1 + Z**-1)
..*;
..*DEF .?QC = 2/TS                          ;THIS IS THE CONSTANT C USED IN THIS M
..*DEF .?QA = - REAL (POLE 1)*TPI
..*DEF .?QB = IMAG (POLE 1)*TPI
..*DEF .?QA0 = (.?QC + .?QA)**2 + .?QB **2
..*DEF .?QA1 = 2*(.?QA**2 + .?QB**2 - .?QC**2)
..*DEF .?QA2 = (.?QC-.?QA)**2 + .?QB**2
..*DEF .?QRADIUS = SQR (.?QA2/.?QA0)
..*DEF .?QANGLE = -.?QA1/ (.?QA0*2*.?QRADIUS)
..*IF .?QANGLE>1
...*.?QANGLE=1
...*ORIF .?QANGLE<-1
...*.?QANGLE=-1
...*END
..*.?QANGLE = ACOS (.?QANGLE)
..*DEF POLE 100 = .?QRADIUS, .?QANGLE, Z
..*DEF ZERO 100 = 1, PI, Z
..*DEF ZERO (100+1) = 1, PI, Z
..*;
..*;ADJUST FOR REAL IF NECESSARY
..*:CKREAL 1,100,POLE,ZERO,.?QC,.?QA0
...*; SUB MACRO USED IN ANOTHER SUB-MACRO BLTCOM TO CHECK FOR REAL POLE/ZERO AND
...*;            ANGLE ETC
...*IF IMAG(POLE 1)=0 THEN
....*       REM ZERO 100+1          ;REMOVE EXTRA PZ
....*       .?QA0=SQR(.?QA0)           ;ADJUST SCALE FACTOR
....*       IF ABS(TPI*REAL(POLE 1))<.?QC THEN    ;ADJUST ANGLE
.....*                  MOV  POLE 100 TO RAD(POLE 100),0
.....*       ELSE
.....*                  MOV  POLE 100 TO RAD(POLE 100),PI
.....*       END
....*END
...*EM
..*;
..*WRITE 'SCALE FACTOR = ', 1/.?QA0
..*;
..*REM .?QC
..*REM .?QA
..*REM .?QB
..*REM .?QA0
..*REM .?QA1
..*REM .?QA2
```

```
. . *REM . ?QRADIUS
. . *REM . ?QANGLE
. . *EM
. *EM
SCALE FACTOR = 1.01007783/10**8
*
*
*:BTP 2,200
. *         ;THIS MACRO GENERATES A BILINEAR TRANSFORM OF A GIVEN POLE
. *         ;CALLING SEQUENCE    :BTP  POLE # IN S, POLE # IN Z
. *         ;EXAMPLE             :BTP  3,90
. *         ;                    THIS WILL TAKE A POLE IN THE S PLANE (P 3)SPECIFIE
. *         ;                    IN HZ, AND PRODUCE A POLE IN THE Z PLANE (P 90)
. *         ;                    AND 1 OR 2 EXTRA ZEROES (Z 90 / Z 91)
. *         ;                    DEPENDING ON WHETHER P 3 IS A REAL POLE OR COMPLEX
. *         ; CALLS SUB-MACRO BLTCOM, CKREAL
. *:BLTCOM 2, 200, POLE, ZERO
. . *; THIS MACRO IS CALLED FROM BTP/BTZ, WHICH SUPPLY POLE/ZERO
. . *; LABELS ,AND THE PROPER 'POLE'/'ZERO'
. . *
. . *; THE TRANSFORM EQUATION HERE IS
. . *;
. . *;            S  = (C) * (1-Z**-1) / (1 + Z**-1)
. . *;
. . *DEF . ?QC = 2/TS                          ;THIS IS THE CONSTANT C USED IN THIS M
. . *DEF . ?QA = - REAL (POLE 2)*TPI
. . *DEF . ?QB = IMAG (POLE 2)*TPI
. . *DEF . ?QA0 = (. ?QC + . ?QA)**2 + . ?QB **2
. . *DEF . ?QA1 = 2*(. ?QA**2 + . ?QB**2 - . ?QC**2)
. . *DEF . ?QA2 = (. ?QC-. ?QA)**2 + . ?QB**2
. . *DEF . ?QRADIUS = SQR (. ?QA2/. ?QA0)
. . *DEF . ?QANGLE = -. ?QA1/ (. ?QA0*2*. ?QRADIUS)
. . *IF . ?QANGLE>1
. . . *. ?QANGLE=1
. . . *ORIF . ?QANGLE<-1
. . . *. ?QANGLE=-1
. . . *END
. . *. ?QANGLE = ACOS (. ?QANGLE)
. . *DEF POLE 200 = . ?QRADIUS, . ?QANGLE, Z
. . *DEF ZERO 200 = 1, PI, Z
. . *DEF ZERO (200+1) = 1, PI, Z
. . *;
. . *;ADJUST FOR REAL IF NECESSARY
. . *:CKREAL 2,200,POLE,ZERO,.?QC,.?QA0
. . . *;  SUB MACRO USED IN ANOTHER SUB-MACRO BLTCOM TO CHECK FOR REAL POLE/ZERO AND
. . . *;             ANGLE ETC
. . . *IF IMAG(POLE 2)=0 THEN
. . . . *       REM ZERO 200+1          ;REMOVE EXTRA PZ
. . . . *       .?QA0=SQR(.?QA0)           ;ADJUST SCALE FACTOR
. . . . *       IF ABS(TPI*REAL(POLE 2))<.?QC THEN    ;ADJUST ANGLE
. . . . . *                 MOV  POLE 200 TO RAD(POLE 200),0
. . . . . *       ELSE
. . . . . *                 MOV  POLE 200 TO RAD(POLE 200),PI
. . . . . *       END
```

```
. . . . *END
. . . *EM
. . *;
. . *WRITE 'SCALE FACTOR = ', 1/.?QA0
. . *;
. . *REM . ?QC
. . *REM . ?QA
. . *REM . ?QB
. . *REM . ?QA0
. . *REM . ?QA1
. . *REM . ?QA2
. . *REM . ?QRADIUS
. . *REM . ?QANGLE
. . *EM
. *EM
1 POLES/ZEROES REMOVED
1 POLES/ZEROES MOVED
SCALE FACTOR = 8.9753904/10**5
*
*
*
*P
POLE 1 = -249.99995,433.01272,CONTINUOUS
POLE 2 = -500.00000,0.00000000,CONTINUOUS;  REAL
POLE 100 = 0.70162724,0.67729125,Z
POLE 200 = 0.43606005,0.00000000,Z;  REAL
*Z
ZERO 100 = 1.00000000,3.1415927,Z;  REAL
ZERO 101 = 1.00000000,3.1415927,Z;  REAL
ZERO 200 = 1.00000000,3.1415927,Z;  REAL
*REMOVE POLE 1
1 POLES/ZEROES REMOVED
*REMOVE POLE 2
1 POLES/ZEROES REMOVED
*
*
```

```
*
*
*
*
*
*
*
*
*
*
*FSCALE
FSCALE=10.0000000,10000.0000
*FSCALE =10,100,200,300,400,500,600,1000,1500
*                                                GRAPH GAIN

GAIN       !...^...!...^....!...^...!...^....!...^...!...^....!.......!.........!
  0.0      ''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
 -2.3                                                 '''-..
 -4.5                                                     ''--..
 -6.8                                                         '--
 -9.0                                                           -
-11.3                                                            '
-13.5                                                             '
-15.8                                                              '
-18.0                                                               '
-20.3                                                                '
-22.5                                                                 '
-24.8                                                                  '
-27.0                                                                   '
-29.3                                                                    '
-31.5                                                                     '
-33.8                                                                      '
-36.1                                                                       '
-39.3                                                                        -
-40.6                                                                         .
-42.8                                                                         '
-45.1                                                                          '
-47.3                                                                          .
    DB\HZ !...^...!...^....!...^...!...^....!...^...!...^....!.......!.........:
*         10 30 100 140  200 240 300 340  400 450 500 540  600    1000     1500
*
*
*
*
*
*
*
*
*
*GAIN(500)
GAIN(500.00000) = -2.7605857
*; THE DISTORTION IS A RESULT OF THE EVALUATION OF CONSTANT C IN MACRO.
```

```
TS
TS = 2.4999995/10**4
*EVA 1/TS
4.0000004*10**3
*
*
*
*
*
*
*
*
*
*
*XSIZE
XSIZE = 79.000000
*                                    GRAPH STEP

STEP       !.....^.....^.....^.....^...^.....^.....^.....!
39.1                .
37.3                 -
35.4                  /___/////////////////////////////////////////////////////
33.6              -
31.8
30.0
28.2
26.4
24.6             /
22.8
20.9
19.1
17.3
15.5
13.7            /
11.9
10.1
 8.3
 6.4
 4.6           /
 2.8
 1.0           .
  AMP\SEC  !.....^.....^.....^.....^...^.....^.....^.....!
*          0.000   0.003   0.005   0.007 0.0087 0.01   0.012   0.014        0.017
*
*
*
*
*
*
*
*
*
*
*
*
```

REFERENCES

1. 2920 Signal Processing Applications Software/Compiler User's Guide. Santa Clara, California: Intel Corporation, 1980.

2. Stanley, W.D. Digital Signal Processing. Reston, Virginia: Reston Publishing Company, 1975.

3. 2920 Analog Signal Processor Design Handbook. Santa Clara, California: Intel Corporation, 1980.

4. 2920 Simulator User's Guide. Santa Clara, California: Intel Corporation, 1979.

5. ISIS.II Credit (CRT Based Text Editor) User's Guide. Santa Clara, California: Intel Corporation, 1979.

6. 2920 Assembly Language Manual. Santa Clara, California: Intel Corporation, 1979.